UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/671,966	09/26/2003	Tina Fay Schneider	FXPL-01044US3	7252

23910   7590   09/25/2008
FLIESLER MEYER LLP
650 CALIFORNIA STREET
14TH FLOOR
SAN FRANCISCO, CA 94108

EXAMINER
CHOI, MICHELE C

ART UNIT	PAPER NUMBER
2169	

MAIL DATE	DELIVERY MODE
09/25/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

	Application No.	Applicant(s)
Office Action Summary	10/671,966	SCHNEIDER ET AL.
	Examiner	**Art Unit**	
	MICHELE C. CHOI	2169	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 June 2008*.

2a)☒ This action is **FINAL**.   2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3,4,6,8-12 and 19-32* is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3,4,6,8-12 and 19-32* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *13 February 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☐ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .

5)☐ Notice of Informal Patent Application

6)☐ Other: _____.

DETAILED ACTION

Response to Amendment

1.   This Office Action is in response to applicant's communication filed June 25,

2008 in response to PTO Office Action mailed April 2, 2008. The Applicant's remarks

and amendments to the claims and/or the specification were considered with the results

that follow.

2.   In response to the last Office Action, claims 1, 3-4, 21, 23-24, and 31-32 have

been amended. Claims 2, 5, 7, and 13-18 have been previously canceled. Claims 6, 8-

12, 19-20, 22, and 25-30 have been previously presented. As a result, claims 1, 3-4, 6,

8-12, and 19-32 are now pending in this application.

Claim Rejections - 35 USC § 112

3.   The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly
> claiming the subject matter which the applicant regards as his invention.

4.   **Claim 3** is rejected under 35 U.S.C. 112, second paragraph, as being indefinite

for failing to particularly point out and distinctly claim the subject matter which applicant

regards as the invention.

        Claim 3 recites the limitation "the same at least one property object". There is

insufficient antecedent basis for this limitation in the claim.

Claim Rejections - 35 USC § 103

5.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6.   **Claims 1, 3-4, 6, 8-12, and 19-32** are rejected under 35 U.S.C. 103(a) as being

unpatentable over Shimizu et al. (US Patent 6,374,271 B1), in view of Newman et al.

(US Patent 6,154, 600), in further view of Rutledge (USPA Pub 2004/0054627 A1), and

in further view of Efrat et al. (US 6,570,587 B1).

        As per **claim 1**, Shimizu discloses, "providing a collection basket tool interface for

configuring program properties, the collection basket tool interface comprising a

program basket workspace" as **[*The hypermedia authoring system...provides a*

display for creating a presentation outline in the form of Bento-boxes. Each

Bento-box includes a layout workspace in which spacer objects may be placed.

Spatial and temporal parameters may also be specified corresponding to each

spacer object. The spacer objects may be linked to cards in the card database

and an indication of a link relationship with the goals outline is also provided

(column 1, line 66 – column 2, line 6); Fig. 13]**. Furthermore, Shimizu discloses, "and

an auxiliary workspace" as **[Fig. 8]**. Furthermore, Shimizu discloses, "the program

basket workspace comprises a plurality of program slots" as **[*The hypermedia*

authoring system...provides a display for creating a presentation outline in the form of Bento-boxes. Each Bento-box includes a layout workspace in which spacer objects may be placed. Spatial and temporal parameters may also be specified corresponding to each spacer object. The spacer objects may be linked to cards in the card database and an indication of a link relationship with the goals outline is also provided (column 1, line 66 – column 2, line 6); Fig. 13].

Furthermore, Shimizu discloses, "generating at least one program that includes one or more program properties, wherein the one or more program properties are associated with the imported media" as **[*The invention provides a hypermedia authoring system that enables an author to generate a document using a goals outline and a presentation outline and provides a technique to relate the goals and presentation outlines. The goals outline organizes the information content of the document contained in cards in a logical structure while the presentation outline is directed to a physical appearance of the document. The goals outline is generated by instantiating document prototypes that contain slots which indicate a type of information that is to be included in the document prototype. Each of the slot types may be linked to the cards by links which may be mandatory, optional, or user defined...Once instantiated, the hypermedia authoring system provides a display that permits the author to link each of the slot types of an instantiated document prototype with cards in a card database. Cards may...be created from scratch by the author or imported from external sources and linked with the instantiated document prototype* (column 1, lines 43-65); *The cards in the***

card database 106 are information elements (called points) on which the

document being authored is based...the cards contain the actual hypermedia

information such as video, sound or text that make up the document. The

contents of the points may be appropriated from either external sources or

created from scratch by the author (column 3, lines 38-44); *The hypermedia*

authoring system also provides a display for creating a presentation outline in

the form of Bento-boxes. Each Bento-box includes a layout workspace in which

spacer objects may be placed. Spatial and temporal parameters may also be

specified corresponding to each spacer object. The spacer objects may be linked

to cards in the card database and an indication of a link relationship with the

goals outline is also provided (column 1, line 66 – column 2, line 6); *The related*

goals outline nodes are goals outline nodes that are linked to cards which are

also linked to spacer objects in the Bento-box 800 (column 7, lines 48-50); **Fig. 8,**

Fig. 10, Fig. 11, and Fig. 13]. Furthermore, Shimizu discloses, "each program is a

program object" as **[***The cards in the card database 106 are information elements*

(called points) on which the document being authored is based...the cards

contain the actual hypermedia information such as video, sound or text that make

up the document. The contents of the points may be appropriated from either

external sources or created from scratch by the author (column 3, lines 38-44);

Bento-box 800 (which may be a card) (column 7, lines 36-37); *The Bento-boxes*

may be linked to each other in a stack for a sequential slide presentation (column

2, lines 12-13); Fig. 13 and Fig. 15]. Furthermore, Shimizu discloses, "each program

property is a property object" as **[*The spacer objects 706-718* (column 7, line 35);**

related goals outline nodes* (column 7, lines 46-48); *The related goals outline

nodes are goals outline nodes that are linked to cards which are also linked to

spacer objects in the Bento-box 800* (column 7, lines 48-50); *The goals outline

organizes the information content of the document contained in cards in a logical

structure...The goals outline is generated by instantiating document prototypes

that contain slots which indicate a type of information that is to be included in the

document prototype. Each of the slot types may be linked to the cards by links

which may be mandatory, optional, or user defined* (column 1, lines 45-55); *Each

Bento-box includes a layout workspace in which spacer objects may be placed.

Spatial and temporal parameters may also be specified corresponding to each

spacer object. The spacer objects may be linked to cards in the card database

and an indication of a link relationship with the goals outline is also provided

(column 2, lines 1-6); Fig. 8, Fig. 10, Fig. 11, and Fig. 13]. Furthermore, Shimizu

discloses, "each program object references at least one property object" as **[*The***

related goals outline nodes are goals outline nodes that are linked to cards which

are also linked to spacer objects in the Bento-box 800* (column 7, lines 48-50); *The

cards in the card database 106 are information elements (called points) on which

the document being authored is based...the cards contain the actual hypermedia

information such as video, sound or text that make up the document. The

contents of the points may be appropriated from either external sources or

***created from scratch by the author* (column 3, lines 38-44); Fig. 8 and Fig. 13]**.

Furthermore, Shimizu discloses, "creating a program element to represent each

generated program, wherein each program element is displayed within a separate

program slot of the program basket workspace" as **[*The hypermedia authoring**

system...provides a display for creating a presentation outline in the form of

Bento-boxes. Each Bento-box includes a layout workspace in which spacer

objects may be placed. Spatial and temporal parameters may also be specified

corresponding to each spacer object. The spacer objects may be linked to cards

in the card database and an indication of a link relationship with the goals outline

is also provided (column 1, line 66 – column 2, line 6); *The cards in the card**

database 106 are information elements (called points) on which the document

being authored is based...the cards contain the actual hypermedia information

such as video, sound or text that make up the document. The contents of the

points may be appropriated from either external sources or created from scratch

by the author (column 3, lines 38-44); *Bento-box 800 (which may be a card)**

(column 7, lines 36-37); *The Bento-boxes may be linked to each other in a stack

for a sequential slide presentation (column 2, lines 12-13); *The cutting board 1508**

displays the presentation outline. The Bento-boxes 1510 and 1512 are shown on

the cutting board 1508 (column 9, lines 39-42); Fig. 13 and Fig. 21]**. Finally, Shimizu

discloses, "editing the programs and properties based on user input obtained through

the collection basket tool interface" as **[*The hypermedia authoring system...provides**

a display for creating a presentation outline in the form of Bento-boxes. Each

Bento-box includes a layout workspace in which spacer objects may be placed.

Spatial and temporal parameters may also be specified corresponding to each

spacer object. The spacer objects may be linked to cards in the card database

and an indication of a link relationship with the goals outline is also provided

(column 1, line 66 – column 2, line 6); the Bento-box 800 also includes a related

goals outline area 806 that shows related goals outline nodes (column 7, lines 45-

47); The goals outline organizes the information content of the document

contained in cards in a logical structure...The goals outline is generated by

instantiating document prototypes that contain slots which indicate a type of

information that is to be included in the document prototype. Each of the slot

types may be linked to the cards by links which may be mandatory, optional, or

user defined (column 1, lines 45-55); The related goals outline nodes are goals

outline nodes that are linked to cards which are also linked to spacer objects in

the Bento-box 800 (column 7, lines 48-50); Fig. 8, Fig. 10, Fig. 11, Fig. 13, and Fig.

15].

Shimizu does not explicitly teach, "a computer-implemented method for

processing media and configuring program properties associated with the media for a

digital document", "the auxiliary workspace comprises at least one media configuration

tool", and "each program property is one of a common program property, a media

related property, a synchronization property, an annotation property...a narration

property and a border property".

However, Newman discloses, "a computer-implemented method for processing

media and configuring program properties associated with the media for a digital

document" as [*The present invention provides a complete, standalone, economical system that enables consumers to create, manipulate, edit, view and output hypermedia (column 17, lines 60-63); The term hypermedia refers to the integration of text, graphics, sound, video, and other data, or any combination into a primarily associative system for information presentation, storage and retrieval. For example, hypermedia includes motion pictures, music, animation and photographs (column 5, lines 47-50); FIG. 12 illustrates an embodiment of a graphics GUI 490. The graphics GUI 490 includes a graphics tab 492 having a scroll bar 494 and a plurality of icons 496 representing various graphics overlays, such as color, titles, and text on color. The graphics GUI 490 similarly includes a display window 408 to display an image for graphical editing. The graphics GUI 490 also includes an image edit window 498 having a plurality of image editing tool buttons 500, such as a line tool, a box tool, a text tool, cut, copy and paste, about the periphery of the image edit window 498...Moreover, the graphics GUI 490 includes a features portion 502. The features portion 502 includes a name box 504, wherein the consumer may select a name to identify the edited graphic and a plurality of tabs 506, each tab having property selections for some of the image editing tool buttons 500. For example, a color tab 508 includes selection boxes 510 to select color features of lines, fills and shadows. Lastly, the features portion 502 includes a pair of slider bars 512, each having a slider 514, for the selection of color gradients. Operation of the graphics editor is similar to conventional graphics editors, such as MacPaint and Paintbrush (column 16, lines 21-44); Fig.*

12]. Furthermore, Newman discloses, "the auxiliary workspace comprises at least one

media configuration tool" as **[*The present invention provides a complete,*

standalone, economical system that enables consumers to create, manipulate,

edit, view and output hypermedia (column 17, lines 60-63); The term hypermedia

refers to the integration of text, graphics, sound, video, and other data, or any

combination into a primarily associative system for information presentation,

storage and retrieval. For example, hypermedia includes motion pictures, music,

animation and photographs (column 5, lines 47-50); FIG. 12 illustrates an

embodiment of a graphics GUI 490. The graphics GUI 490 includes a graphics tab

492 having a scroll bar 494 and a plurality of icons 496 representing various

graphics overlays, such as color, titles, and text on color. The graphics GUI 490

similarly includes a display window 408 to display an image for graphical editing.

The graphics GUI 490 also includes an image edit window 498 having a plurality

of image editing tool buttons 500, such as a line tool, a box tool, a text tool, cut,

copy and paste, about the periphery of the image edit window 498...Moreover, the

graphics GUI 490 includes a features portion 502. The features portion 502

includes a name box 504, wherein the consumer may select a name to identify the

edited graphic and a plurality of tabs 506, each tab having property selections for

some of the image editing tool buttons 500. For example, a color tab 508 includes

selection boxes 510 to select color features of lines, fills and shadows. Lastly, the

features portion 502 includes a pair of slider bars 512, each having a slider 514,

for the selection of color gradients. Operation of the graphics editor is similar to

conventional graphics editors, such as MacPaint and Paintbrush **(column 16,**

lines 21-44); Fig. 12]. Finally, Newman discloses, "each program property is one of a

common program property, a media related property, a synchronization property, an

annotation property,…a narration property and a border property" as **[*The present**

invention provides a complete, standalone, economical system that enables

consumers to create, manipulate, edit, view and output hypermedia* (column 17,

lines 60-63); *The term hypermedia refers to the integration of text, graphics,*

sound, video, and other data, or any combination into a primarily associative

system for information presentation, storage and retrieval. For example,

hypermedia includes motion pictures, music, animation and photographs

(column 5, lines 47-50); *Referring now to FIG. 8, a flowchart illustrates a method*

for manipulating hypermedia. At state 380, a consumer captures hypermedia to a

storyboard. The storyboard often includes a plurality of captured hypermedia

portions, or clips. At state 382, the consumer manipulates the captured

hypermedia on the storyboard…For example, consumers manipulate clips by

enhancing them with graphics, text and audio annotations. Consumers may

likewise enhance electronic mail (e-mail) by excerpting, editing and attaching the

edited clips to their e-mail messages. Consumers also manipulate clips by

compiling a sequence of clips and creating the transitions between each pair of

clips. Additionally, consumers may manipulate clips by incorporating digitized

photographs, synchronized music and other forms of digital hypermedia

captured from the Internet via the Internet browser functionality incorporated into

the non-linear editing system 100. For example, a consumer may manipulate a

clip by excerpting a video still or a digital photograph from a clip for placement

on a World Wide Web page or for printout as a photograph or postcard. At state

384, the consumer may modify the default transitions selected by the non-linear

editing system 100. Moreover, at state 386, the system 100 enables consumers to

add overlays, such as graphical and audio annotations of clips. Lastly,

consumers may play back their storyboards at state 388 or copy their

storyboards at state 390 (column 14, lines 10-37); Fig. 8].

Shimizu and Newman are analogous art because they both teach a method for

configuring properties for a digital document.

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to have combined the teachings of Newman with the teachings of

Shimizu because Newman's teachings provide Shimizu's method with a complete,

standalone, economical system that enables consumers to create, manipulate, edit,

view and output hypermedia **[column 17, lines 60-63]**. For example, the graphics GUI

allows a user to edit graphics included in a hypermedia **[column 16, lines 21-44]**.

Furthermore, Shimizu and Newman both do not explicitly teach, "receiving user

input to select the media to import using the media configuration tool" and "importing the

media selected by the user".

However, Rutledge discloses, "receiving user input to select the media to import

using the media configuration tool" as **[*The media ID software has a sophisticated*

graphical-user-interface (or GUI) that allows the user to navigate through the

locally stored media IDs, initiate complicated search queries for local and remote

media and media IDs, and retrieve articles of media and media IDs... the media ID

and the media ID software contain the necessary information required to transfer

the entire text and graphics of the original article of media between the web

server and the media ID software 1080...when the user clicks on a media ID such

as the one located at 1070 in FIG. 22, the entire text and graphics of the media

article 1068 will be transferred along with the media ID 1070 to the media ID

software 1080 **(page 27, paragraph [0275], lines 1-24);** *The media ID software*

program manages media IDs and their content, provides advanced query tools for

searching and sorting of media ID content, manages the download and transfer of

media IDs and associated electronic documents, and includes screens, windows

or other facilities for quick and easy previewing of electronic documents **(page**

31, paragraph [0319], lines 1-7); Fig. 33, Fig. 37 and Fig. 39-41]. Finally, Rutledge

discloses, "importing the media selected by the user" as **[***The media ID software has a*

sophisticated graphical-user-interface (or GUI) that allows the user to navigate

through the locally stored media IDs, initiate complicated search queries for local

and remote media and media IDs, and retrieve articles of media and media IDs...

the media ID and the media ID software contain the necessary information

required to transfer the entire text and graphics of the original article of media

between the web server and the media ID software 1080...when the user clicks on

a media ID such as the one located at 1070 in FIG. 22, the entire text and graphics

of the media article 1068 will be transferred along with the media ID 1070 to the

media ID software 1080 **(page 27, paragraph [0275], lines 1-24);** *The media ID*

software program manages media IDs and their content, provides advanced

query tools for searching and sorting of media ID content, manages the download

and transfer of media IDs and associated electronic documents, and includes

screens, windows or other facilities for quick and easy previewing of electronic

documents **(page 31, paragraph [0319], lines 1-7); Fig. 33, Fig. 37 and Fig. 39-41]**.

Shimizu, Newman, and Rutledge are analogous art because all three teach a

method of configuring hypermedia.

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to have combined the teachings of Rutledge with the combined

teachings of Shimizu and Newman because Rutledge's teachings provide the

combination of Shimizu's and Newman's method with sophisticated media search tools

to facilitate the media search process **[(page 36, paragraph [0369], lines 1-2); Fig. 39**

- Fig. 41].

Shimizu, Newman, and Rutledge all do not explicitly teach, "each program

property is one of…a hotspot property".

However, Efrat discloses, "each program property is one of…a hotspot property"

as **[***The authoring tool 1001 may be used to define hotspots in media, such as*

video, and link 1010 the hotspots to targets **(column 5, lines 2-3)]**.

Shimizu, Newman, Rutledge, and Efrat are analogous art because all three teach

a method of configuring hypermedia.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined the teachings of Efrat with the combined teachings of Shimizu, Newman, and Rutledge because Efrat's teachings provide the combination of Shimizu's, Newman's, and Rutledge's method with the ability to create and display hyperlinks, or hotspots, in select regions of interest in frames of a video **[column 2, lines 15-17]**.

As per **claim 3**, Shimizu discloses, "a first program object and a second program object reference the same at least one property object" as **[*the caption spacer object 710 is highlighted as indicated by the dotted box 802 and the corresponding card C is highlighted as indicated by the dotted box 803. If a spacer object 706-718 is linked to more than one card, then all the cards linked to the selected spacer object are highlighted* (column 7, lines 40-45); Fig. 13]**.

As per **claim 4**, Shimizu discloses, "a first program object and a second program object reference duplicate property objects" as **[*the caption spacer object 710 is highlighted as indicated by the dotted box 802 and the corresponding card C is highlighted as indicated by the dotted box 803. If a spacer object 706-718 is linked to more than one card, then all the cards linked to the selected spacer object are highlighted* (column 7, lines 40-45); Fig. 13]**.

As per **claim 6**, Shimizu discloses, "exporting the program from the program basket workspace to a stage channel of the digital document" as **[*The hypermedia authoring system...provides a display for creating a presentation outline in the form of Bento-boxes. Each Bento-box includes a layout workspace in which spacer objects may be placed. Spatial and temporal parameters may also be specified corresponding to each spacer object. The spacer objects may be linked to cards in the card database and an indication of a link relationship with the goals outline is also provided* (column 1, line 66 – column 2, line 6); *the hypermedia authoring system provides a display that permits the author to link each of the slot types of an instantiated document prototype with the cards in a card database. Cards may...be created from scratch by the author or imported from external sources and linked with the instantiated document prototype* (column 1, lines 63-65); *Bento-box 800 (which may be a card)*; *The cutting board 1508 displays the presentation outline. The Bento-boxes 1510 and 1512 are shown on the cutting board 1508* (column 9, lines 39-42); Fig. 13 and Fig. 21]**.

As per **claim 8**, Shimizu discloses, "implementing the collection basket tool interface as a graphic user interface, wherein the program basket workspace is implemented as a first graphical user interface window" as **[*The hypermedia authoring system...provides a display for creating a presentation outline in the form of Bento-boxes. Each Bento-box includes a layout workspace in which spacer objects may be placed. Spatial and temporal parameters may also be specified***

corresponding to each spacer object. The spacer objects may be linked to cards in the card database and an indication of a link relationship with the goals outline is also provided **(column 1, line 66 – column 2, line 6); Fig. 13]**. Furthermore, Shimizu discloses, "the auxiliary workspace is implemented as a second graphical user interface window" as **[*The goals outline processor 608 processes the goals outline by displaying a goals outline display 400 on the display device 605 and generates the goals outline 102 based on user inputs received through the keyboard/mouse 609...Fig. 8 shows the goals outline display 400 that includes a tree view area 402, a card link area 404, and a document prototype legend area 406. The tree view 402 displays the tree view 408 of the goals outline 202. Each of the nodes 206-214 are represented by square boxes that are connected to a respective icon such as icon 432 corresponding to node 206. Each of the icons 432, 436, 438, 440, and 460 are coded, by color for example, based on the document prototype that is instantiated to correspond to the respective nodes 206-214. For example, the icon 432 is colored blue as represented by the horizontal parallel lines; icon 438 is colored red as represented by the parallel vertical lines; and icon 440 is colored green as represented by the parallel slanted lines. As shown in the legend area 406, the blue color indicated in area 424 corresponds to a description document prototype; the green color indicated in area 426 corresponds to an argument prototype; and the red color indicated in area 428 corresponds to a narrative prototype* (column 5, lines 44-65); Fig. 8]**.

As per **claim 9**, Rutledge discloses, "providing the auxiliary workspace includes:

providing a media search tool" as **[*The media ID software has a sophisticated

graphical-user-interface (or GUI) that allows the user to navigate through the

locally stored media IDs, initiate complicated search queries for local and remote

media and media IDs, and retrieve articles of media and media IDs* (page 27, lines

1-5); *The media ID software program...provides advanced query tools for

searching and sorting of media ID content* (page 31, paragraph [0319], lines 1-3);

*The media ID software contains sophisticated media and media ID search tools as

well as sophisticated media and media ID filtering tools. A media ID search

engine window 1622 in FIG. 39 provides powerful and flexible search capabilities.

The media ID search engines window 1622 contains a search matrix 1624 that the

user can utilize to design and execute searches for media IDs and associated

media* (page 36, paragraph [0369], lines 1-8); Fig. 39 and Fig. 41]**.

As per **claim 10**, Shimizu discloses, "providing the auxiliary workspace includes:

providing a scene configuration tool, said scene configuration tool used to configure one

or more scenes included in one or more programs" as **[*The presentation outline

processor...provides support for the author to generate a presentation

outline...FIG. 10 shows a layout workspace 700 where the author may generate a

physical appearance of the document spatially as well as temporally by placing

spacer objects in a spatial layout area 702 and spacer objects in a sound layout

area 704. For example, the author may place visual spacer objects 706-712 and*

sound spacer objects 714, 716 and 718 as shown in FIG. 10. Each of the spacer

objects may be time sequenced using a temporal view 810 of the layout

workspace 700 as shown in FIG. 11...The spacer objects 706-718 in the layout

workspace 700 are linked to corresponding cards by using a Bento-box 800

(which may be a card) which include the layout workspace as shown in Fig. 13. A

card link area 804 shows the cards that are linked to the corresponding spacer

objects 706-718...In addition to the card link area 804, the Bento-Box 800 also

includes a related goals outline area 806 that shows related goals outline nodes.

The related goals outline nodes are goals outline nodes that are linked to cards

which are also linked to spacer objects in the Bento-box 800...The Bento-box 800

provides a method for linking the presentation outline to the cards in the card

database 106 which indirectly also links the presentation outline 104 with the

goals outline 102 as indicated by the related goals outline area 806 **(column 7,**

lines 16-59); Fig. 10 - Fig. 13].

As per **claim 11**, Shimizu discloses, "providing the auxiliary workspace includes:

providing a slide configuration tool, said slide configuration tool used to configure a slide

show of programs" as **[***The Bento-boxes may be linked to each other in a stack for*

a sequential slide presentation **(column 2, lines 12-13);** *Fig. 14 shows a stack of*

Bento-boxes 830, 840 and 850. Bento-box 830 titled Bento 1 is displayed first and

then Bento-box 840 titled Bento 2 is displayed second and then Bento-box 850

titled Bento 3 is displayed third...Bento-boxes 830, 840 and 850 are linked to each

other in a stack to indicate a serial presentation of the information contained in

each of the Bento-boxes 830, 840 and 850. A Bento-box stack may be given titles

so that each of the Bento-box stacks may be referenced as a whole when linked

to other Bento-boxes or Bento-box stacks. For example, FIG. 15 shows such a

sequence of Bento-boxes slide #1 902, slide #2 904 and slide #3 906. These

Bento-boxes 902-906 are arranged sequentially in a stack so that Bento-boxes

902, 904 and 906 are presented in sequential order as a slideshow presentation

(column 7, line 60 – column 8, line 7); Fig. 14 - Fig. 15].

As per **claim 12**, Rutledge discloses, "the media search tool includes a media

viewer to allow a user to preview a search result" as **[*The media ID software program*

manages media IDs and their content, provides advanced query tools for

searching and sorting of media ID content, manages the download and transfer of

media IDs and associated electronic documents, and includes screens, windows

or other facilities for quick and easy previewing of electronic documents (page**

31, paragraph [0319], lines 1-7); *A media search results window 1632 shows the*

results from a media ID search executed from the media search engine 1622. In

this example, three search results were returned in the media search results

window 1632. The user of the media ID software can click on one or more of these

search results to access the media ID associated with it and or the original article

of media **(page 36, paragraph [0370], lines 1-7); Fig. 33 and Fig. 39]**.

As per **claim 19**, Newman discloses, "at least one of the property objects

reference one or more methods" as **[the storyboard GUI 440 includes a button panel**

458 having a plurality of buttons corresponding to a wide variety of editing

functions for the storyboard 446, such as save, print, cut, copy and paste (column

15, lines 29-32); Fig. 10].

As per **claim 20**, Rutledge discloses, "providing a media filter to analyze the

media before importing the media to ensure that the media's format is supported by the

collection basket tool interface" as **[The media ID software contains sophisticated**

media and media ID search tools as well as sophisticated media and media ID

filtering tools (page 36, paragraph [0369], lines 1-3); The media ID software has a

sophisticated graphical-user-interface (or GUI) that allows the user to navigate

through the locally stored media IDs, initiate complicated search queries for local

and remote media and media IDs, and retrieve articles of media and media IDs.

This graphical-user-interface 1086 provides functionality for the user to accept

the default media ID organization structure or to configure a custom media ID

organization structure...Each media ID has complete information about the media

article such as publisher, date published, number of pages, media type, author,

title, icon type, etc. In many cases the electronic media ID and the original media

are stored together (the entire text, graphics, sound, video, etc) in the media ID

software. In other embodiments, the media ID and the media ID software contain

the necessary information required to transfer the entire text and graphics of the

original article of media between the web server and the media ID software 1080.

In one embodiment of the present invention, when the user clicks on a media ID

such as the one located at 1070 in FIG. 22, the entire text and graphics of the

media article 1068 will be transferred along with the media ID 1070 to the media ID

software 1080 (page 27, paragraph [0275], lines 1-24); *The media ID software*

program manages media IDs and their content, provides advanced query tools for

searching and sorting of media ID content, manages the download and transfer of

media IDs and associated electronic documents, and includes screens, windows

or other facilities for quick and easy previewing of electronic documents (page

31, paragraph [0319], lines 1-7); Fig. 33, Fig. 37, and Fig. 39-41].

As per **claim 21**, Shimizu discloses, "creating one or more property icons

included with each program element, wherein the property icons represent the program

properties referenced by the program associated with the displayed program element"

as [*The hypermedia authoring system...provides a display for creating a*

presentation outline in the form of Bento-boxes. Each Bento-box includes a

layout workspace in which spacer objects may be placed. Spatial and temporal

parameters may also be specified corresponding to each spacer object. The

spacer objects may be linked to cards in the card database and an indication of a

link relationship with the goals outline is also provided (column 1, line 66 –

column 2, line 6); Fig. 13]. Finally, Shimizu discloses, "configuring programs by

manipulating program properties through actions performed on their associated property

icons within the program basket workspace, wherein the actions are performed based

on user input" as **[*The goals outline processor 608 processes the goals outline by*

displaying a goals outline display 400 on the display device 605 and generates

the goals outline 102 based on user inputs received through the keyboard/mouse

609...Fig. 8 shows the goals outline display 400 that includes a tree view area 402,

a card link area 404, and a document prototype legend area 406. The tree view 402

displays the tree view 408 of the goals outline 202. Each of the nodes 206-214 are

represented by square boxes that are connected to a respective icon such as

icon 432 corresponding to node 206. Each of the icons 432, 436, 438, 440, and 460

are coded, by color for example, based on the document prototype that is

instantiated to correspond to the respective nodes 206-214. For example, the icon

432 is colored blue as represented by the horizontal parallel lines; icon 438 is

colored red as represented by the parallel vertical lines; and icon 440 is colored

green as represented by the parallel slanted lines. As shown in the legend area

406, the blue color indicated in area 424 corresponds to a description document

prototype; the green color indicated in area 426 corresponds to an argument

prototype; and the red color indicated in area 428 corresponds to a narrative

prototype (column 5, lines 44-65); Fig. 8]**.

As per **claim 22**, Shimizu discloses, "the actions performed on the property icons

comprise: deleting property icons" as [***The inherit column 518 indicates whether the

link type for each slot name is inherited from a higher level document prototype.

Because document prototypes may be recursively defined (i.e., a document

prototype being defined by other document prototypes), the link types of higher

level document prototypes may either be applied to the lower level document

prototypes such as indicated by symbol 514 corresponding to the Process slot

name or be overwritten by the lower level document prototype as indicated by

symbol 516 corresponding to the Historical slot name. Thus, the browser display

500 provides the author complete flexibility in defining document prototypes and

when used in conjunction with the goals outline display 400 as shown in Fig. 8,

and each of the instantiated document prototypes may be linked to a card in the

card database 106 (column 7, lines 1-15); Fig. 8; See 516 in Fig. 9**]. Furthermore,

Shimizu discloses, "copying property icons from one program slot to another program

slot" as [***the caption spacer object 710 is highlighted as indicated by the dotted

box 802 and the corresponding card C is highlighted as indicated by the dotted

box 803. If a spacer object 706-718 is linked to more than one card, then all the

cards linked to the selected spacer object are highlighted (column 7, lines 40-45);

The inherit column 518 indicates whether the link type for each slot name is

inherited from a higher level document prototype. Because document prototypes

may be recursively defined (i.e., a document prototype being defined by other

document prototypes), the link types of higher level document prototypes may**

either be applied to the lower level document prototypes such as indicated by

symbol 514 corresponding to the Process slot name or be overwritten by the

lower level document prototype as indicated by symbol 516 corresponding to the

Historical slot name. Thus, the browser display 500 provides the author complete

flexibility in defining document prototypes and when used in conjunction with the

goals outline display 400 as shown in Fig. 8, and each of the instantiated

document prototypes may be linked to a card in the card database 106 **(column 7,**

lines 1-15); Fig. 8; See 514 in Fig. 9; Fig. 13]. Finally, Shimizu discloses, "moving

property icons from one program slot to another program slot" as **[*the caption spacer***

object 710 is highlighted as indicated by the dotted box 802 and the

corresponding card C is highlighted as indicated by the dotted box 803. If a

spacer object 706-718 is linked to more than one card, then all the cards linked to

the selected spacer object are highlighted **(column 7, lines 40-45);** *The inherit*

column 518 indicates whether the link type for each slot name is inherited from a

higher level document prototype. Because document prototypes may be

recursively defined (i.e., a document prototype being defined by other document

prototypes), the link types of higher level document prototypes may either be

applied to the lower level document prototypes such as indicated by symbol 514

corresponding to the Process slot name or be overwritten by the lower level

document prototype as indicated by symbol 516 corresponding to the Historical

slot name. Thus, the browser display 500 provides the author complete flexibility

in defining document prototypes and when used in conjunction with the goals

outline display 400 as shown in Fig. 8, and each of the instantiated document

prototypes may be linked to a card in the card database 106 (column 7, lines 1-

15); Fig. 8; See 514 in Fig. 9; Fig. 13].

As per **claim 23**, Newman discloses, "receiving user input to select a first

property icon" as [See *Graphics tab 492* in Fig. 12]. Furthermore, Newman discloses,

"generating a property editor tool that corresponds to a property associated with the first

property icon" as [Fig. 12]. Furthermore, Newman discloses, "the property editor tool is

capable of configuring common program properties, media related properties,

synchronization properties, annotation properties…narration properties and border

properties" as [*The present invention provides a complete, standalone, economical*

system that enables consumers to create, manipulate, edit, view and output

hypermedia (column 17, lines 60-63); *The term hypermedia refers to the*

integration of text, graphics, sound, video, and other data, or any combination

into a primarily associative system for information presentation, storage and

retrieval. For example, hypermedia includes motion pictures, music, animation

and photographs (column 5, lines 47-50); *Referring now to FIG. 8, a flowchart*

illustrates a method for manipulating hypermedia. At state 380, a consumer

captures hypermedia to a storyboard. The storyboard often includes a plurality of

captured hypermedia portions, or clips. At state 382, the consumer manipulates

the captured hypermedia on the storyboard…For example, consumers

manipulate clips by enhancing them with graphics, text and audio annotations.

Consumers may likewise enhance electronic mail (e-mail) by excerpting, editing

and attaching the edited clips to their e-mail messages. Consumers also

manipulate clips by compiling a sequence of clips and creating the transitions

between each pair of clips. Additionally, consumers may manipulate clips by

incorporating digitized photographs, synchronized music and other forms of

digital hypermedia captured from the Internet via the Internet browser

functionality incorporated into the non-linear editing system 100. For example, a

consumer may manipulate a clip by excerpting a video still or a digital

photograph from a clip for placement on a World Wide Web page or for printout

as a photograph or postcard. At state 384, the consumer may modify the default

transitions selected by the non-linear editing system 100. Moreover, at state 386,

the system 100 enables consumers to add overlays, such as graphical and audio

annotations of clips. Lastly, consumers may play back their storyboards at state

388 or copy their storyboards at state 390 **(column 14, lines 10-37); Fig. 8].**

Furthermore, Efrat discloses, "the property editor tool is capable of configuring…hotspot

properties" as **[*The authoring tool 1001 may be used to define hotspots in media,***

***such as video, and link 1010 the hotspots to targets* (column 5, lines 2-3)].** Finally,

Newman discloses, "editing at least one element of the selected property based on user

input obtained through the property editor tool" as **[*FIG. 12 illustrates an embodiment***

of a graphics GUI 490. The graphics GUI 490 includes a graphics tab 492 having a

scroll bar 494 and a plurality of icons 496 representing various graphics overlays,

such as color, titles, and text on color. The graphics GUI 490 similarly includes a

display window 408 to display an image for graphical editing. The graphics GUI

490 also includes an image edit window 498 having a plurality of image editing

tool buttons 500, such as a line tool, a box tool, a text tool, cut, copy and paste,

about the periphery of the image edit window 498...Moreover, the graphics GUI

490 includes a features portion 502. The features portion 502 includes an name

box 504, wherein the consumer may select a name to identify the edited graphic

and a plurality of tabs 506, each tab having property selections for some of the

image editing tool buttons 500. For example, a color tab 508 includes selection

boxes 510 to select color features of lines, fills and shadows. Lastly, the features

portion 502 includes a pair of slider bars 512, each having a slider 514, for the

selection of color gradients. Operation of the graphics editor is similar to

conventional graphics editors, such as MacPaint and Paintbrush **(column 16,**

lines 21-44); Fig. 12].

As per **claim 24**, Newman discloses, "editing the program and property objects

comprises: receiving user input to select a first program element within the program

basket workspace" as **[See** *"Intro Title" graphic selected in Graphics Tab in* **Fig. 12;**

The graphics GUI 490...includes a display window 408 to display an image for

graphical editing **(column 16, lines 25-26); Fig. 12]**. Furthermore, Newman discloses,

"generating a property editor tool that corresponds to the program associated with the

selected program element, wherein the property editor tool identifies the program

properties referenced by the program associated with the selected program element" as

[*FIG. 12 illustrates an embodiment of a graphics GUI 490. The graphics GUI 490*

includes a graphics tab 492 having a scroll bar 494 and a plurality of icons 496

representing various graphics overlays, such as color, titles, and text on color.

The graphics GUI 490 similarly includes a display window 408 to display an image

for graphical editing. The graphics GUI 490 also includes an image edit window

498 having a plurality of image editing tool buttons 500, such as a line tool, a box

tool, a text tool, cut, copy and paste, about the periphery of the image edit

window 498...Moreover, the graphics GUI 490 includes a features portion 502. The

features portion 502 includes an name box 504, wherein the consumer may select

a name to identify the edited graphic and a plurality of tabs 506, each tab having

property selections for some of the image editing tool buttons 500. For example,

a color tab 508 includes selection boxes 510 to select color features of lines, fills

and shadows. Lastly, the features portion 502 includes a pair of slider bars 512,

each having a slider 514, for the selection of color gradients. Operation of the

graphics editor is similar to conventional graphics editors, such as MacPaint and

Paintbrush (column 16, lines 21-44); Fig. 12]. Furthermore, Newman discloses, "the

property editor tool is capable of editing common program properties, media related

properties, synchronization properties, annotation properties...narration properties and

border properties" as [*The present invention provides a complete, standalone,*

economical system that enables consumers to create, manipulate, edit, view and

output hypermedia (column 17, lines 60-63); *The term hypermedia refers to the*

integration of text, graphics, sound, video, and other data, or any combination

into a primarily associative system for information presentation, storage and

retrieval. For example, hypermedia includes motion pictures, music, animation

and photographs (column 5, lines 47-50); Referring now to FIG. 8, a flowchart

illustrates a method for manipulating hypermedia. At state 380, a consumer

captures hypermedia to a storyboard. The storyboard often includes a plurality of

captured hypermedia portions, or clips. At state 382, the consumer manipulates

the captured hypermedia on the storyboard...For example, consumers

manipulate clips by enhancing them with graphics, text and audio annotations.

Consumers may likewise enhance electronic mail (e-mail) by excerpting, editing

and attaching the edited clips to their e-mail messages. Consumers also

manipulate clips by compiling a sequence of clips and creating the transitions

between each pair of clips. Additionally, consumers may manipulate clips by

incorporating digitized photographs, synchronized music and other forms of

digital hypermedia captured from the Internet via the Internet browser

functionality incorporated into the non-linear editing system 100. For example, a

consumer may manipulate a clip by excerpting a video still or a digital

photograph from a clip for placement on a World Wide Web page or for printout

as a photograph or postcard. At state 384, the consumer may modify the default

transitions selected by the non-linear editing system 100. Moreover, at state 386,

the system 100 enables consumers to add overlays, such as graphical and audio

annotations of clips. Lastly, consumers may play back their storyboards at state

388 or copy their storyboards at state 390 (column 14, lines 10-37); Fig. 8].

Furthermore, Efrat discloses, "the property editor tool is capable of configuring…hotspot

properties" as **[The authoring tool 1001 may be used to define hotspots in media,**

such as video, and link 1010 the hotspots to targets (column 5, lines 2-3)].

Furthermore, Newman discloses, "receiving user input to select one of the program

properties identified by the property editor tool" as **[See Graphics tab 492 in Fig. 12]**.

Finally, Newman discloses, "editing the selected program property based on user input

obtained through the property editor tool" as **[FIG. 12 illustrates an embodiment of a**

graphics GUI 490. The graphics GUI 490 includes a graphics tab 492 having a

scroll bar 494 and a plurality of icons 496 representing various graphics overlays,

such as color, titles, and text on color. The graphics GUI 490 similarly includes a

display window 408 to display an image for graphical editing. The graphics GUI

490 also includes an image edit window 498 having a plurality of image editing

tool buttons 500, such as a line tool, a box tool, a text tool, cut, copy and paste,

about the periphery of the image edit window 498…Moreover, the graphics GUI

490 includes a features portion 502. The features portion 502 includes an name

box 504, wherein the consumer may select a name to identify the edited graphic

and a plurality of tabs 506, each tab having property selections for some of the

image editing tool buttons 500. For example, a color tab 508 includes selection

boxes 510 to select color features of lines, fills and shadows. Lastly, the features

portion 502 includes a pair of slider bars 512, each having a slider 514, for the

selection of color gradients. Operation of the graphics editor is similar to

conventional graphics editors, such as MacPaint and Paintbrush (column 16,

lines 21-44); Fig. 12].

As per **claim 25**, Newman discloses, "implementing the property editor tool as a

graphical user interface window which appears as a separate interface, wherein the

property editor tool includes property tabs to identify the program properties referenced

by the program associated with the selected program element" as **[*FIG. 12 illustrates*

an embodiment of a graphics GUI 490. The graphics GUI 490 includes a graphics

tab 492 having a scroll bar 494 and a plurality of icons 496 representing various

graphics overlays, such as color, titles, and text on color. The graphics GUI 490

similarly includes a display window 408 to display an image for graphical editing.

The graphics GUI 490 also includes an image edit window 498 having a plurality

of image editing tool buttons 500, such as a line tool, a box tool, a text tool, cut,

copy and paste, about the periphery of the image edit window 498...Moreover, the

graphics GUI 490 includes a features portion 502. The features portion 502

includes an name box 504, wherein the consumer may select a name to identify

the edited graphic and a plurality of tabs 506, each tab having property selections

for some of the image editing tool buttons 500. For example, a color tab 508

includes selection boxes 510 to select color features of lines, fills and shadows.

Lastly, the features portion 502 includes a pair of slider bars 512, each having a

slider 514, for the selection of color gradients. Operation of the graphics editor is

similar to conventional graphics editors, such as MacPaint and Paintbrush
(column 16, lines 21-44); Fig. 12].

As per **claim 26**, Rutledge discloses, "the common program properties include a

program name, a unique identifier, a user defined tag, a program description and

references to other properties" as [*The media ID software has a sophisticated*

graphical-user-interface (or GUI) that...provides functionality for the user to

accept the default media ID organization structure or to configure a custom media

ID organization structure...Each media ID has complete information about the

media article such as publisher, date published, number of pages, media type,

author, title, icon type, etc. **(page 27, paragraph [0275], lines 1-13) Fig. 37]**.

As per **claim 27**, Rutledge discloses, "the media related properties include

attributes applicable to an individual media type, and other media related properties

including author information, creation and modified dates, and media copyright

information" as [*Each media ID has complete information about the media article*

such as publisher, date published, number of pages, media type, author, title,

icon type, etc. **(page 27, paragraph [0275], lines 10-13); Fig. 37]**.

As per **claim 28**, Efrat discloses, "the hotspot properties include hotspot shape,

size, location, action, text and highlighting" as **[*The authoring tool 1001 may be used*

to define hotspots in media, such as video, and link 1010 the hotspots to targets

(column 5, lines 2-3); *The user can create a hypervideo that is nonlinear 1009 by*

linking 1010 a hotspot in a source media to one or more targets 7003. A target

7003 may be a playing instance of a media element 1690 **(column 18, lines 23-26);**

The Hotspot Properties...may include the hotspot name 7103, type of shape 7105

defining the hotspot, media file name in which the hotspot is located 7107, the

range of frames in a video media file in which the hotspot 8003 is located 7109,

and cursor types. The cursor types include the cursor displayed when the pointer

is over the hotspot 7111, and the cursor displayed when the mouse button is

actuated when the pointer is over the hotspot 7113 **(column 12, lines 55-64);** *size*

of the hotspot 8003 **(column 17, line 12);** *hotspots in the form of an identifier (ID)*

(i.e. text) **(column 6, lines 60-61); Fig. 7C]**.

As per **claim 29**, Newman discloses, "the narration properties include font

properties and text display related attributes" as **[*consumers manipulate clips by*

enhancing them with graphics, text, and audio annotations **(column 14, lines 18-**

19); See *Font Style* **in Fig. 12]**.

As per **claim 30**, Newman discloses, "the border properties include border text, size, colors and fonts" as [*The graphics GUI 490...includes an image edit window 498 having a plurality of image editing tool buttons 500, such as a line tool, a box tool, a text tool, cut, copy and paste, about the periphery of the image edit window 498...Moreover, the graphics GUI 490 includes a features portion 502. The features portion 502 includes...a plurality of tabs 506, each tab having property selections for some of the image editing tool buttons 500. For example, a color tab 508 includes selection boxes 510 to select color features of lines, fills and shadows. Lastly, the features portion 502 includes a pair of slider bars 512, each having a slider 514, for the selection of color gradients* (column 16, lines 26-42); See *Color tab 508* and *Line and Font Style tabs 506* in Fig. 12].

As per **claim 31**, Shimizu discloses, "providing a collection basket tool interface for configuring program properties, the collection basket tool interface comprising a program basket workspace" as [*The hypermedia authoring system...provides a display for creating a presentation outline in the form of Bento-boxes. Each Bento-box includes a layout workspace in which spacer objects may be placed. Spatial and temporal parameters may also be specified corresponding to each spacer object. The spacer objects may be linked to cards in the card database and an indication of a link relationship with the goals outline is also provided* (column 1, line 66 – column 2, line 6); Fig. 13]. Furthermore, Shimizu discloses, "an auxiliary workspace" as [Fig. 8]. Furthermore, Shimizu discloses, "the program basket

workspace comprises a plurality of program slots" as **[The hypermedia authoring**

system...provides a display for creating a presentation outline in the form of

Bento-boxes. Each Bento-box includes a layout workspace in which spacer

objects may be placed. Spatial and temporal parameters may also be specified

corresponding to each spacer object. The spacer objects may be linked to cards

in the card database and an indication of a link relationship with the goals outline

is also provided (column 1, line 66 – column 2, line 6); Fig. 13]. Furthermore,

Shimizu discloses, "implementing the collection basket tool interface as a graphic user

interface, wherein the program basket workspace is implemented as a first graphical

user interface window" as **[The hypermedia authoring system...provides a display**

for creating a presentation outline in the form of Bento-boxes. Each Bento-box

includes a layout workspace in which spacer objects may be placed. Spatial and

temporal parameters may also be specified corresponding to each spacer object.

The spacer objects may be linked to cards in the card database and an indication

of a link relationship with the goals outline is also provided (column 1, line 66 –

column 2, line 6); Fig. 13]. Furthermore, Shimizu discloses, "the auxiliary workspace is

implemented as a second graphical user interface window" as **[The goals outline**

processor 608 processes the goals outline by displaying a goals outline display

400 on the display device 605 and generates the goals outline 102 based on user

inputs received through the keyboard/mouse 609...Fig. 8 shows the goals outline

display 400 that includes a tree view area 402, a card link area 404, and a

document prototype legend area 406. The tree view 402 displays the tree view 408

of the goals outline 202. Each of the nodes 206-214 are represented by square

boxes that are connected to a respective icon such as icon 432 corresponding to

node 206. Each of the icons 432, 436, 438, 440, and 460 are coded, by color for

example, based on the document prototype that is instantiated to correspond to

the respective nodes 206-214. For example, the icon 432 is colored blue as

represented by the horizontal parallel lines; icon 438 is colored red as

represented by the parallel vertical lines; and icon 440 is colored green as

represented by the parallel slanted lines. As shown in the legend area 406, the

blue color indicated in area 424 corresponds to a description document

prototype; the green color indicated in area 426 corresponds to an argument

prototype; and the red color indicated in area 428 corresponds to a narrative

prototype **(column 5, lines 44-65); Fig. 8]**. Furthermore, Shimizu discloses,

"generating at least one program that includes one or more program properties, wherein

the one or more program properties are associated with the imported media" as **[*The*

invention provides a hypermedia authoring system that enables an author to

generate a document using a goals outline and a presentation outline and

provides a technique to relate the goals and presentation outlines. The goals

outline organizes the information content of the document contained in cards in a

logical structure while the presentation outline is directed to a physical

appearance of the document. The goals outline is generated by instantiating

document prototypes that contain slots which indicate a type of information that

is to be included in the document prototype. Each of the slot types may be linked

to the cards by links which may be mandatory, optional, or user defined...Once

instantiated, the hypermedia authoring system provides a display that permits the

author to link each of the slot types of an instantiated document prototype with

cards in a card database. Cards may...be created from scratch by the author or

imported from external sources and linked with the instantiated document

prototype **(column 1, lines 43-65);** *The cards in the card database 106 are*

information elements (called points) on which the document being authored is

based...the cards contain the actual hypermedia information such as video,

sound or text that make up the document. The contents of the points may be

appropriated from either external sources or created from scratch by the author

(column 3, lines 38-44); *The hypermedia authoring system also provides a display*

for creating a presentation outline in the form of Bento-boxes. Each Bento-box

includes a layout workspace in which spacer objects may be placed. Spatial and

temporal parameters may also be specified corresponding to each spacer object.

The spacer objects may be linked to cards in the card database and an indication

of a link relationship with the goals outline is also provided **(column 1, line 66 –**

column 2, line 6); *The related goals outline nodes are goals outline nodes that are*

linked to cards which are also linked to spacer objects in the Bento-box 800

(column 7, lines 48-50); Fig. 8, Fig. 10, Fig. 11, and Fig. 13]. Furthermore, Shimizu

discloses, "each program is a program object" as **[***The cards in the card database 106*

are information elements (called points) on which the document being authored

is based...the cards contain the actual hypermedia information such as video,

sound or text that make up the document. The contents of the points may be

appropriated from either external sources or created from scratch by the author

(column 3, lines 38-44); *Bento-box 800 (which may be a card)* **(column 7, lines 36-**

37); *The Bento-boxes may be linked to each other in a stack for a sequential slide*

presentation **(column 2, lines 12-13); Fig. 13 and Fig. 15]**. Furthermore, Shimizu

discloses, "each program property is a property object" as **[***The spacer objects 706-*

718 **(column 7, line 35);** *related goals outline nodes* **(column 7, lines 46-48);** *The*

related goals outline nodes are goals outline nodes that are linked to cards which

are also linked to spacer objects in the Bento-box 800 **(column 7, lines 48-50);** *The*

goals outline organizes the information content of the document contained in

cards in a logical structure...The goals outline is generated by instantiating

document prototypes that contain slots which indicate a type of information that

is to be included in the document prototype. Each of the slot types may be linked

to the cards by links which may be mandatory, optional, or user defined **(column**

1, lines 45-55); *Each Bento-box includes a layout workspace in which spacer*

objects may be placed. Spatial and temporal parameters may also be specified

corresponding to each spacer object. The spacer objects may be linked to cards

in the card database and an indication of a link relationship with the goals outline

is also provided **(column 2, lines 1-6); Fig. 8, Fig. 10, Fig. 11, and Fig. 13]**.

Furthermore, Shimizu discloses, "each program object references at least one property

object" as **[***The related goals outline nodes are goals outline nodes that are linked*

to cards which are also linked to spacer objects in the Bento-box 800 **(column 7,**

lines 48-50); *The cards in the card database 106 are information elements (called*

points) on which the document being authored is based...the cards contain the

actual hypermedia information such as video, sound or text that make up the

document. The contents of the points may be appropriated from either external

sources or created from scratch by the author **(column 3, lines 38-44); Fig. 8 and**

Fig. 13]. Furthermore, Shimizu discloses, "creating a program element to represent

each generated program, wherein each program element is displayed within a separate

program slot of the program basket workspace" as **[***The hypermedia authoring*

system...provides a display for creating a presentation outline in the form of

Bento-boxes. Each Bento-box includes a layout workspace in which spacer

objects may be placed. Spatial and temporal parameters may also be specified

corresponding to each spacer object. The spacer objects may be linked to cards

in the card database and an indication of a link relationship with the goals outline

is also provided **(column 1, line 66 – column 2, line 6);** *The cards in the card*

database 106 are information elements (called points) on which the document

being authored is based...the cards contain the actual hypermedia information

such as video, sound or text that make up the document. The contents of the

points may be appropriated from either external sources or created from scratch

by the author **(column 3, lines 38-44);** *Bento-box 800 (which may be a card)*

(column 7, lines 36-37); *The Bento-boxes may be linked to each other in a stack*

for a sequential slide presentation **(column 2, lines 12-13);** *The cutting board 1508*

displays the presentation outline. The Bento-boxes 1510 and 1512 are shown on

the cutting board 1508 (column 9, lines 39-42); Fig. 13 and Fig. 21]. Furthermore,

Shimizu discloses, "creating one or more property icons included with each program

element, wherein the property icons represent the program properties referenced by the

program associated with the displayed program element **[The hypermedia authoring**

system...provides a display for creating a presentation outline in the form of

Bento-boxes. Each Bento-box includes a layout workspace in which spacer

objects may be placed. Spatial and temporal parameters may also be specified

corresponding to each spacer object. The spacer objects may be linked to cards

in the card database and an indication of a link relationship with the goals outline

is also provided (column 1, line 66 – column 2, line 6); Fig. 13]. Finally, Shimizu

discloses, "configuring the programs by manipulating the program properties through

actions performed on their associated property icons within the program basket

workspace, wherein the actions are performed based on user input" as **[The goals**

outline processor 608 processes the goals outline by displaying a goals outline

display 400 on the display device 605 and generates the goals outline 102 based

on user inputs received through the keyboard/mouse 609...Fig. 8 shows the goals

outline display 400 that includes a tree view area 402, a card link area 404, and a

document prototype legend area 406. The tree view 402 displays the tree view 408

of the goals outline 202. Each of the nodes 206-214 are represented by square

boxes that are connected to a respective icon such as icon 432 corresponding to

node 206. Each of the icons 432, 436, 438, 440, and 460 are coded, by color for

example, based on the document prototype that is instantiated to correspond to

the respective nodes 206-214. For example, the icon 432 is colored blue as

represented by the horizontal parallel lines; icon 438 is colored red as

represented by the parallel vertical lines; and icon 440 is colored green as

represented by the parallel slanted lines. As shown in the legend area 406, the

blue color indicated in area 424 corresponds to a description document

prototype; the green color indicated in area 426 corresponds to an argument

prototype; and the red color indicated in area 428 corresponds to a narrative

prototype (column 5, lines 44-65); Fig. 8].

Shimizu does not explicitly teach, "a computer-implemented method for

processing media and configuring program properties associated with the media for a

digital document", "a property editor tool", "the auxiliary workspace comprises at least

one media configuration tool", "the property editor tool is capable of editing common

program properties, media related properties, synchronization properties, annotation

properties...narration properties and border properties", "the property editor tool is

implemented as a third graphical user interface window", "configuring the program

properties, further comprising the steps of: receiving user input to select a first property

icon", "generating the property editor tool that corresponds to the property associated

with the first property icon", and "editing at least one element of the selected property

based on user input obtained through the property editor tool".

However, Newman discloses, "a computer-implemented method for processing

media and configuring program properties associated with the media for a digital

document" as **[The present invention provides a complete, standalone, economical**

system that enables consumers to create, manipulate, edit, view and output

hypermedia (column 17, lines 60-63); *The term hypermedia refers to the*

integration of text, graphics, sound, video, and other data, or any combination

into a primarily associative system for information presentation, storage and

retrieval. For example, hypermedia includes motion pictures, music, animation

and photographs (column 5, lines 47-50); *FIG. 12 illustrates an embodiment of a*

graphics GUI 490. The graphics GUI 490 includes a graphics tab 492 having a

scroll bar 494 and a plurality of icons 496 representing various graphics overlays,

such as color, titles, and text on color. The graphics GUI 490 similarly includes a

display window 408 to display an image for graphical editing. The graphics GUI

490 also includes an image edit window 498 having a plurality of image editing

tool buttons 500, such as a line tool, a box tool, a text tool, cut, copy and paste,

about the periphery of the image edit window 498...Moreover, the graphics GUI

490 includes a features portion 502. The features portion 502 includes a name box

504, wherein the consumer may select a name to identify the edited graphic and a

plurality of tabs 506, each tab having property selections for some of the image

editing tool buttons 500. For example, a color tab 508 includes selection boxes

510 to select color features of lines, fills and shadows. Lastly, the features portion

502 includes a pair of slider bars 512, each having a slider 514, for the selection

of color gradients. Operation of the graphics editor is similar to conventional

graphics editors, such as MacPaint and Paintbrush (column 16, lines 21-44); **Fig.**

12]. Furthermore, Newman discloses, "a property editor tool" as **[Fig. 12]**. Furthermore,

Newman discloses, "the auxiliary workspace comprises at least one media configuration

tool" as **[The present invention provides a complete, standalone, economical**

system that enables consumers to create, manipulate, edit, view and output

hypermedia (column 17, lines 60-63); The term hypermedia refers to the

integration of text, graphics, sound, video, and other data, or any combination

into a primarily associative system for information presentation, storage and

retrieval. For example, hypermedia includes motion pictures, music, animation

and photographs (column 5, lines 47-50); FIG. 12 illustrates an embodiment of a

graphics GUI 490. The graphics GUI 490 includes a graphics tab 492 having a

scroll bar 494 and a plurality of icons 496 representing various graphics overlays,

such as color, titles, and text on color. The graphics GUI 490 similarly includes a

display window 408 to display an image for graphical editing. The graphics GUI

490 also includes an image edit window 498 having a plurality of image editing

tool buttons 500, such as a line tool, a box tool, a text tool, cut, copy and paste,

about the periphery of the image edit window 498...Moreover, the graphics GUI

490 includes a features portion 502. The features portion 502 includes a name box

504, wherein the consumer may select a name to identify the edited graphic and a

plurality of tabs 506, each tab having property selections for some of the image

editing tool buttons 500. For example, a color tab 508 includes selection boxes

510 to select color features of lines, fills and shadows. Lastly, the features portion

502 includes a pair of slider bars 512, each having a slider 514, for the selection

of color gradients. Operation of the graphics editor is similar to conventional

graphics editors, such as MacPaint and Paintbrush **(column 16, lines 21-44); Fig.**

12]. Furthermore, Newman discloses, "the property editor tool is capable of editing

common program properties, media related properties, synchronization properties,

annotation properties…narration properties and border properties" as **[*The present***

invention provides a complete, standalone, economical system that enables

***consumers to create, manipulate, edit, view and output hypermedia* (column 17,**

lines 60-63); *The term hypermedia refers to the integration of text, graphics,*

sound, video, and other data, or any combination into a primarily associative

system for information presentation, storage and retrieval. For example,

hypermedia includes motion pictures, music, animation and photographs

(column 5, lines 47-50); *Referring now to FIG. 8, a flowchart illustrates a method*

for manipulating hypermedia. At state 380, a consumer captures hypermedia to a

storyboard. The storyboard often includes a plurality of captured hypermedia

portions, or clips. At state 382, the consumer manipulates the captured

hypermedia on the storyboard…For example, consumers manipulate clips by

enhancing them with graphics, text and audio annotations. Consumers may

likewise enhance electronic mail (e-mail) by excerpting, editing and attaching the

edited clips to their e-mail messages. Consumers also manipulate clips by

compiling a sequence of clips and creating the transitions between each pair of

clips. Additionally, consumers may manipulate clips by incorporating digitized

photographs, synchronized music and other forms of digital hypermedia

captured from the Internet via the Internet browser functionality incorporated into

the non-linear editing system 100. For example, a consumer may manipulate a

clip by excerpting a video still or a digital photograph from a clip for placement

on a World Wide Web page or for printout as a photograph or postcard. At state

384, the consumer may modify the default transitions selected by the non-linear

editing system 100. Moreover, at state 386, the system 100 enables consumers to

add overlays, such as graphical and audio annotations of clips. Lastly,

consumers may play back their storyboards at state 388 or copy their

storyboards at state 390 **(column 14, lines 10-37); Fig. 8]**. Furthermore, Newman

discloses, "the property editor tool is implemented as a third graphical user interface

window" as **[See Fig. 12]**. Furthermore, Newman discloses, "configuring the program

properties, further comprising the steps of: receiving user input to select a first property

icon" as **[See *Graphics tab 492* in Fig. 12]**. Furthermore, Newman discloses,

"generating the property editor tool that corresponds to the property associated with the

first property icon" as **[Fig. 12]**. Finally, Newman discloses, "editing at least one element

of the selected property based on user input obtained through the property editor tool"

as **[*FIG. 12 illustrates an embodiment of a graphics GUI 490. The graphics GUI 490*

includes a graphics tab 492 having a scroll bar 494 and a plurality of icons 496

representing various graphics overlays, such as color, titles, and text on color.

The graphics GUI 490 similarly includes a display window 408 to display an image

for graphical editing. The graphics GUI 490 also includes an image edit window

498 having a plurality of image editing tool buttons 500, such as a line tool, a box

tool, a text tool, cut, copy and paste, about the periphery of the image edit

window 498…Moreover, the graphics GUI 490 includes a features portion 502. The features portion 502 includes an name box 504, wherein the consumer may select a name to identify the edited graphic and a plurality of tabs 506, each tab having property selections for some of the image editing tool buttons 500. For example, a color tab 508 includes selection boxes 510 to select color features of lines, fills and shadows. Lastly, the features portion 502 includes a pair of slider bars 512, each having a slider 514, for the selection of color gradients. Operation of the graphics editor is similar to conventional graphics editors, such as MacPaint and Paintbrush (column 16, lines 21-44); Fig. 12].*

Shimizu and Newman are analogous art because they both teach a method for configuring properties for a digital document.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined the teachings of Newman with the teachings of Shimizu because Newman's teachings provide Shimizu's method with a complete, standalone, economical system that enables consumers to create, manipulate, edit, view and output hypermedia **[column 17, lines 60-63]**. For example, the graphics GUI allows a user to edit graphics included in a hypermedia **[column 16, lines 21-44]**.

Furthermore, Shimizu and Newman both do not explicitly teach, "receiving user input to select the media to import using the media configuration tool" and "importing the media selected by the user".

However, Rutledge discloses, "receiving user input to select the media to import using the media configuration tool" as **[*The media ID software has a sophisticated***

graphical-user-interface (or GUI) that allows the user to navigate through the

locally stored media IDs, initiate complicated search queries for local and remote

media and media IDs, and retrieve articles of media and media IDs... the media ID

and the media ID software contain the necessary information required to transfer

the entire text and graphics of the original article of media between the web

server and the media ID software 1080...when the user clicks on a media ID such

as the one located at 1070 in FIG. 22, the entire text and graphics of the media

article 1068 will be transferred along with the media ID 1070 to the media ID

software 1080 (page 27, paragraph [0275], lines 1-24); *The media ID software*

program manages media IDs and their content, provides advanced query tools for

searching and sorting of media ID content, manages the download and transfer of

media IDs and associated electronic documents, and includes screens, windows

or other facilities for quick and easy previewing of electronic documents (page

31, paragraph [0319], lines 1-7); Fig. 33, Fig. 37 and Fig. 39-41]. Finally, Rutledge

discloses, "importing the media selected by the user" as **[*The media ID software has a*

sophisticated graphical-user-interface (or GUI) that allows the user to navigate

through the locally stored media IDs, initiate complicated search queries for local

and remote media and media IDs, and retrieve articles of media and media IDs...

the media ID and the media ID software contain the necessary information

required to transfer the entire text and graphics of the original article of media

between the web server and the media ID software 1080...when the user clicks on

a media ID such as the one located at 1070 in FIG. 22, the entire text and graphics

of the media article 1068 will be transferred along with the media ID 1070 to the

media ID software 1080 (page 27, paragraph [0275], lines 1-24); *The media ID*

software program manages media IDs and their content, provides advanced

query tools for searching and sorting of media ID content, manages the download

and transfer of media IDs and associated electronic documents, and includes

screens, windows or other facilities for quick and easy previewing of electronic

documents (page 31, paragraph [0319], lines 1-7); Fig. 33, Fig. 37 and Fig. 39-41].

Shimizu, Newman, and Rutledge are analogous art because all three teach a

method of configuring hypermedia.

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to have combined the teachings of Rutledge with the combined

teachings of Shimizu and Newman because Rutledge's teachings provide the

combination of Shimizu's and Newman's method with sophisticated media search tools

to facilitate the media search process **[(page 36, paragraph [0369], lines 1-2); Fig. 39**

- Fig. 41].

Furthermore, Shimizu, Newman, and Rutledge all do not explicitly teach, "the

property editor tool is capable of configuring...hotspot properties".

However, Efrat discloses, "the property editor tool is capable of

configuring...hotspot properties" as *[The authoring tool 1001 may be used to define*

hotspots in media, such as video, and link 1010 the hotspots to targets (column 5,

lines 2-3)].

Shimizu, Newman, Rutledge, and Efrat are analogous art because all three teach

a method of configuring hypermedia.

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to have combined the teachings of Efrat with the combined

teachings of Shimizu, Newman, and Rutledge because Efrat's teachings provide the

combination of Shimizu's, Newman's, and Rutledge's method with the ability to create

and display hyperlinks, or hotspots, in select regions of interest in frames of a video

[column 2, lines 15-17].

As per **<u>claim 32</u>**, Shimizu discloses, "providing a collection basket tool interface

for configuring program properties, the collection basket tool interface comprising a

program basket workspace" as **[*The hypermedia authoring system…provides a*

display for creating a presentation outline in the form of Bento-boxes. Each

Bento-box includes a layout workspace in which spacer objects may be placed.

Spatial and temporal parameters may also be specified corresponding to each

spacer object. The spacer objects may be linked to cards in the card database

and an indication of a link relationship with the goals outline is also provided

(column 1, line 66 – column 2, line 6); Fig. 13]. Furthermore, Shimizu discloses, "an

auxiliary workspace" as **[Fig. 8]**. Furthermore, Shimizu discloses, "the program basket

workspace comprises a plurality of program slots" as **[*The hypermedia authoring*

system…provides a display for creating a presentation outline in the form of

Bento-boxes. Each Bento-box includes a layout workspace in which spacer

objects may be placed. Spatial and temporal parameters may also be specified

corresponding to each spacer object. The spacer objects may be linked to cards

in the card database and an indication of a link relationship with the goals outline

is also provided **(column 1, line 66 – column 2, line 6); Fig. 13]**. Furthermore,

Shimizu discloses, "implementing the collection basket tool interface as a graphic user

interface, wherein the program basket workspace is implemented as a first graphical

user interface window" as **[*The hypermedia authoring system...provides a display**

for creating a presentation outline in the form of Bento-boxes. Each Bento-box

includes a layout workspace in which spacer objects may be placed. Spatial and

temporal parameters may also be specified corresponding to each spacer object.

The spacer objects may be linked to cards in the card database and an indication

of a link relationship with the goals outline is also provided **(column 1, line 66 –**

column 2, line 6); Fig. 13]. Furthermore, Shimizu discloses, "the auxiliary workspace is

implemented as a second graphical user interface window" as **[*The goals outline**

processor 608 processes the goals outline by displaying a goals outline display

400 on the display device 605 and generates the goals outline 102 based on user

inputs received through the keyboard/mouse 609...Fig. 8 shows the goals outline

display 400 that includes a tree view area 402, a card link area 404, and a

document prototype legend area 406. The tree view 402 displays the tree view 408

of the goals outline 202. Each of the nodes 206-214 are represented by square

boxes that are connected to a respective icon such as icon 432 corresponding to

node 206. Each of the icons 432, 436, 438, 440, and 460 are coded, by color for

example, based on the document prototype that is instantiated to correspond to

the respective nodes 206-214. For example, the icon 432 is colored blue as

represented by the horizontal parallel lines; icon 438 is colored red as

represented by the parallel vertical lines; and icon 440 is colored green as

represented by the parallel slanted lines. As shown in the legend area 406, the

blue color indicated in area 424 corresponds to a description document

prototype; the green color indicated in area 426 corresponds to an argument

prototype; and the red color indicated in area 428 corresponds to a narrative

prototype **(column 5, lines 44-65); Fig. 8]**. Furthermore, Shimizu discloses,

"generating at least one program that includes one or more program properties, wherein

the one or more program properties are associated with the imported media" as **[*The*

invention provides a hypermedia authoring system that enables an author to

generate a document using a goals outline and a presentation outline and

provides a technique to relate the goals and presentation outlines. The goals

outline organizes the information content of the document contained in cards in a

logical structure while the presentation outline is directed to a physical

appearance of the document. The goals outline is generated by instantiating

document prototypes that contain slots which indicate a type of information that

is to be included in the document prototype. Each of the slot types may be linked

to the cards by links which may be mandatory, optional, or user defined...Once

instantiated, the hypermedia authoring system provides a display that permits the

author to link each of the slot types of an instantiated document prototype with

cards in a card database. Cards may...be created from scratch by the author or

imported from external sources and linked with the instantiated document

prototype (column 1, lines 43-65); *The cards in the card database 106 are*

information elements (called points) on which the document being authored is

based...the cards contain the actual hypermedia information such as video,

sound or text that make up the document. The contents of the points may be

appropriated from either external sources or created from scratch by the author

(column 3, lines 38-44); *The hypermedia authoring system also provides a display*

for creating a presentation outline in the form of Bento-boxes. Each Bento-box

includes a layout workspace in which spacer objects may be placed. Spatial and

temporal parameters may also be specified corresponding to each spacer object.

The spacer objects may be linked to cards in the card database and an indication

of a link relationship with the goals outline is also provided (column 1, line 66 –

column 2, line 6); *The related goals outline nodes are goals outline nodes that are*

linked to cards which are also linked to spacer objects in the Bento-box 800

(column 7, lines 48-50); Fig. 8, Fig. 10, Fig. 11, and Fig. 13]. Furthermore, Shimizu

discloses, "each program is a program object" as *[The cards in the card database 106*

are information elements (called points) on which the document being authored

is based...the cards contain the actual hypermedia information such as video,

sound or text that make up the document. The contents of the points may be

appropriated from either external sources or created from scratch by the author

(column 3, lines 38-44); *Bento-box 800 (which may be a card)* (column 7, lines 36-

37); *The Bento-boxes may be linked to each other in a stack for a sequential slide presentation* (column 2, lines 12-13); Fig. 13 and Fig. 15]. Furthermore, Shimizu discloses, "each program property is a property object" as [*The spacer objects 706-718* (column 7, line 35); *related goals outline nodes* (column 7, lines 46-48); *The related goals outline nodes are goals outline nodes that are linked to cards which are also linked to spacer objects in the Bento-box 800* (column 7, lines 48-50); *The goals outline organizes the information content of the document contained in cards in a logical structure...The goals outline is generated by instantiating document prototypes that contain slots which indicate a type of information that is to be included in the document prototype. Each of the slot types may be linked to the cards by links which may be mandatory, optional, or user defined* (column 1, lines 45-55); *Each Bento-box includes a layout workspace in which spacer objects may be placed. Spatial and temporal parameters may also be specified corresponding to each spacer object. The spacer objects may be linked to cards in the card database and an indication of a link relationship with the goals outline is also provided* (column 2, lines 1-6); Fig. 8, Fig. 10, Fig. 11, and Fig. 13].

Furthermore, Shimizu discloses, "each program object references at least one property object" as [*The related goals outline nodes are goals outline nodes that are linked to cards which are also linked to spacer objects in the Bento-box 800* (column 7, lines 48-50); *The cards in the card database 106 are information elements (called points) on which the document being authored is based...the cards contain the actual hypermedia information such as video, sound or text that make up the*

document. The contents of the points may be appropriated from either external

sources or created from scratch by the author (column 3, lines 38-44); Fig. 8 and

Fig. 13]. Finally, Shimizu discloses, "creating a program element to represent each

generated program, wherein each program element is displayed within a separate

program slot of the program basket workspace" as [*The hypermedia authoring*

system...provides a display for creating a presentation outline in the form of

Bento-boxes. Each Bento-box includes a layout workspace in which spacer

objects may be placed. Spatial and temporal parameters may also be specified

corresponding to each spacer object. The spacer objects may be linked to cards

in the card database and an indication of a link relationship with the goals outline

is also provided (column 1, line 66 – column 2, line 6); *The cards in the card*

database 106 are information elements (called points) on which the document

being authored is based...the cards contain the actual hypermedia information

such as video, sound or text that make up the document. The contents of the

points may be appropriated from either external sources or created from scratch

by the author (column 3, lines 38-44); *Bento-box 800 (which may be a card)*

(column 7, lines 36-37); *The Bento-boxes may be linked to each other in a stack*

for a sequential slide presentation (column 2, lines 12-13); *The cutting board 1508*

displays the presentation outline. The Bento-boxes 1510 and 1512 are shown on

the cutting board 1508 (column 9, lines 39-42); Fig. 13 and Fig. 21].

Shimizu does not explicitly teach, "a computer-implemented method for

processing media and configuring program properties associated with the media for a

digital document", "a property editor tool", "the auxiliary workspace comprises at least

one media configuration tool", "the property editor tool is capable of editing common

program properties, media related properties, synchronization properties, annotation

properties...narration properties and border properties", "receiving user input to select a

first program element within the program basket workspace", "generating a property

editor tool that corresponds to the program associated with the first program element",

"the property editor tool is implemented as a graphical user interface window which

appears as a separate interface, wherein the property editor tool includes property tabs

to identify the program properties referenced by the program associated with the

selected program element", "receiving user input to select one of the property tabs

associated with the program property", "bringing to the foreground a page within the

property editor tool for configuring the respective program property", and "editing the

respective program property based on user input obtained within the page of the

property editor tool"

However, Newman discloses, "a computer-implemented method for processing

media and configuring program properties associated with the media for a digital

document" as **[The present invention provides a complete, standalone, economical**

system that enables consumers to create, manipulate, edit, view and output

hypermedia (column 17, lines 60-63); The term hypermedia refers to the

integration of text, graphics, sound, video, and other data, or any combination

into a primarily associative system for information presentation, storage and

retrieval. For example, hypermedia includes motion pictures, music, animation

and photographs **(column 5, lines 47-50);** *FIG. 12 illustrates an embodiment of a*

graphics GUI 490. The graphics GUI 490 includes a graphics tab 492 having a

scroll bar 494 and a plurality of icons 496 representing various graphics overlays,

such as color, titles, and text on color. The graphics GUI 490 similarly includes a

display window 408 to display an image for graphical editing. The graphics GUI

490 also includes an image edit window 498 having a plurality of image editing

tool buttons 500, such as a line tool, a box tool, a text tool, cut, copy and paste,

about the periphery of the image edit window 498...Moreover, the graphics GUI

490 includes a features portion 502. The features portion 502 includes a name box

504, wherein the consumer may select a name to identify the edited graphic and a

plurality of tabs 506, each tab having property selections for some of the image

editing tool buttons 500. For example, a color tab 508 includes selection boxes

510 to select color features of lines, fills and shadows. Lastly, the features portion

502 includes a pair of slider bars 512, each having a slider 514, for the selection

of color gradients. Operation of the graphics editor is similar to conventional

graphics editors, such as MacPaint and Paintbrush **(column 16, lines 21-44); Fig.**

12]. Furthermore, Newman discloses, "a property editor tool" as **[Fig. 12]**. Furthermore,

Newman discloses, "the auxiliary workspace comprises at least one media configuration

tool" as **[***The present invention provides a complete, standalone, economical*

system that enables consumers to create, manipulate, edit, view and output

hypermedia (column 17, lines 60-63); The term hypermedia refers to the

integration of text, graphics, sound, video, and other data, or any combination

into a primarily associative system for information presentation, storage and

retrieval. For example, hypermedia includes motion pictures, music, animation

and photographs (column 5, lines 47-50); FIG. 12 illustrates an embodiment of a

graphics GUI 490. The graphics GUI 490 includes a graphics tab 492 having a

scroll bar 494 and a plurality of icons 496 representing various graphics overlays,

such as color, titles, and text on color. The graphics GUI 490 similarly includes a

display window 408 to display an image for graphical editing. The graphics GUI

490 also includes an image edit window 498 having a plurality of image editing

tool buttons 500, such as a line tool, a box tool, a text tool, cut, copy and paste,

about the periphery of the image edit window 498...Moreover, the graphics GUI

490 includes a features portion 502. The features portion 502 includes a name box

504, wherein the consumer may select a name to identify the edited graphic and a

plurality of tabs 506, each tab having property selections for some of the image

editing tool buttons 500. For example, a color tab 508 includes selection boxes

510 to select color features of lines, fills and shadows. Lastly, the features portion

502 includes a pair of slider bars 512, each having a slider 514, for the selection

of color gradients. Operation of the graphics editor is similar to conventional

graphics editors, such as MacPaint and Paintbrush (column 16, lines 21-44); Fig.

12]. Furthermore, Newman discloses, "the property editor tool is capable of editing

common program properties, media related properties, synchronization properties,

annotation properties...narration properties and border properties" as **[*The present*

invention provides a complete, standalone, economical system that enables

consumers to create, manipulate, edit, view and output hypermedia (column 17,

lines 60-63); *The term hypermedia refers to the integration of text, graphics,*

sound, video, and other data, or any combination into a primarily associative

system for information presentation, storage and retrieval. For example,

hypermedia includes motion pictures, music, animation and photographs

(column 5, lines 47-50); *Referring now to FIG. 8, a flowchart illustrates a method*

for manipulating hypermedia. At state 380, a consumer captures hypermedia to a

storyboard. The storyboard often includes a plurality of captured hypermedia

portions, or clips. At state 382, the consumer manipulates the captured

hypermedia on the storyboard...For example, consumers manipulate clips by

enhancing them with graphics, text and audio annotations. Consumers may

likewise enhance electronic mail (e-mail) by excerpting, editing and attaching the

edited clips to their e-mail messages. Consumers also manipulate clips by

compiling a sequence of clips and creating the transitions between each pair of

clips. Additionally, consumers may manipulate clips by incorporating digitized

photographs, synchronized music and other forms of digital hypermedia

captured from the Internet via the Internet browser functionality incorporated into

the non-linear editing system 100. For example, a consumer may manipulate a

clip by excerpting a video still or a digital photograph from a clip for placement

on a World Wide Web page or for printout as a photograph or postcard. At state

384, the consumer may modify the default transitions selected by the non-linear

editing system 100. Moreover, at state 386, the system 100 enables consumers to

add overlays, such as graphical and audio annotations of clips. Lastly,

consumers may play back their storyboards at state 388 or copy their

storyboards at state 390 **(column 14, lines 10-37); Fig. 8]**. Furthermore, Newman

discloses, "receiving user input to select a first program element within the program

basket workspace" as **[See** *"Intro Title" graphic selected in Graphics Tab in* **Fig. 12;**

The graphics GUI 490...includes a display window 408 to display an image for

graphical editing **(column 16, lines 25-26); Fig. 12]**. Furthermore, Newman discloses,

"generating a property editor tool that corresponds to the program associated with the

first program element" as **[Fig. 12]**. Furthermore, Newman discloses, "the property

editor tool is implemented as a graphical user interface window which appears as a

separate interface, wherein the property editor tool includes property tabs to identify the

program properties referenced by the program associated with the selected program

element" as **[***FIG. 12 illustrates an embodiment of a graphics GUI 490. The*

graphics GUI 490 includes a graphics tab 492 having a scroll bar 494 and a

plurality of icons 496 representing various graphics overlays, such as color,

titles, and text on color. The graphics GUI 490 similarly includes a display window

408 to display an image for graphical editing. The graphics GUI 490 also includes

an image edit window 498 having a plurality of image editing tool buttons 500,

such as a line tool, a box tool, a text tool, cut, copy and paste, about the

periphery of the image edit window 498...Moreover, the graphics GUI 490

includes a features portion 502. The features portion 502 includes an name box

504, wherein the consumer may select a name to identify the edited graphic and a

plurality of tabs 506, each tab having property selections for some of the image

editing tool buttons 500. For example, a color tab 508 includes selection boxes

510 to select color features of lines, fills and shadows. Lastly, the features portion

502 includes a pair of slider bars 512, each having a slider 514, for the selection

of color gradients. Operation of the graphics editor is similar to conventional

graphics editors, such as MacPaint and Paintbrush **(column 16, lines 21-44); Fig.**

12]. Furthermore, Newman discloses, "receiving user input to select one of the property

tabs associated with the program property" as **[See *Graphics tab 492* in Fig. 12]**.

Furthermore, Newman discloses, "bringing to the foreground a page within the property

editor tool for configuring the respective program property" as **[See *Graphics tab 492***

in Fig. 12]. Finally, Newman discloses, "editing the respective program property based

on user input obtained within the page of the property editor tool" as **[*FIG. 12 illustrates***

an embodiment of a graphics GUI 490. The graphics GUI 490 includes a graphics

tab 492 having a scroll bar 494 and a plurality of icons 496 representing various

graphics overlays, such as color, titles, and text on color. The graphics GUI 490

similarly includes a display window 408 to display an image for graphical editing.

The graphics GUI 490 also includes an image edit window 498 having a plurality

of image editing tool buttons 500, such as a line tool, a box tool, a text tool, cut,

copy and paste, about the periphery of the image edit window 498...Moreover, the

graphics GUI 490 includes a features portion 502. The features portion 502

includes an name box 504, wherein the consumer may select a name to identify

the edited graphic and a plurality of tabs 506, each tab having property selections

for some of the image editing tool buttons 500. For example, a color tab 508

includes selection boxes 510 to select color features of lines, fills and shadows.

Lastly, the features portion 502 includes a pair of slider bars 512, each having a

slider 514, for the selection of color gradients. Operation of the graphics editor is

similar to conventional graphics editors, such as MacPaint and Paintbrush

(column 16, lines 21-44); Fig. 12].

Shimizu and Newman are analogous art because they both teach a method for

configuring properties for a digital document.

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to have combined the teachings of Newman with the teachings of

Shimizu because Newman's teachings provide Shimizu's method with a complete,

standalone, economical system that enables consumers to create, manipulate, edit,

view and output hypermedia **[column 17, lines 60-63]**. For example, the graphics GUI

allows a user to edit graphics included in a hypermedia **[column 16, lines 21-44]**.

Furthermore, Shimizu and Newman both do not explicitly teach, "receiving user

input to select the media to import using the media configuration tool" and "importing the

media selected by the user".

However, Rutledge discloses, "receiving user input to select the media to import

using the media configuration tool" as **[*The media ID software has a sophisticated*

graphical-user-interface (or GUI) that allows the user to navigate through the

locally stored media IDs, initiate complicated search queries for local and remote

media and media IDs, and retrieve articles of media and media IDs... the media ID

and the media ID software contain the necessary information required to transfer

the entire text and graphics of the original article of media between the web

server and the media ID software 1080...when the user clicks on a media ID such

as the one located at 1070 in FIG. 22, the entire text and graphics of the media

article 1068 will be transferred along with the media ID 1070 to the media ID

software 1080 (page 27, paragraph [0275], lines 1-24); *The media ID software*

program manages media IDs and their content, provides advanced query tools for

searching and sorting of media ID content, manages the download and transfer of

media IDs and associated electronic documents, and includes screens, windows

or other facilities for quick and easy previewing of electronic documents (page

31, paragraph [0319], lines 1-7); Fig. 33, Fig. 37 and Fig. 39-41]. Finally, Rutledge

discloses, "importing the media selected by the user" as [*The media ID software has a*

sophisticated graphical-user-interface (or GUI) that allows the user to navigate

through the locally stored media IDs, initiate complicated search queries for local

and remote media and media IDs, and retrieve articles of media and media IDs...

the media ID and the media ID software contain the necessary information

required to transfer the entire text and graphics of the original article of media

between the web server and the media ID software 1080...when the user clicks on

a media ID such as the one located at 1070 in FIG. 22, the entire text and graphics

of the media article 1068 will be transferred along with the media ID 1070 to the

media ID software 1080 (page 27, paragraph [0275], lines 1-24); *The media ID*

software program manages media IDs and their content, provides advanced

query tools for searching and sorting of media ID content, manages the download

and transfer of media IDs and associated electronic documents, and includes

screens, windows or other facilities for quick and easy previewing of electronic

documents (page 31, paragraph [0319], lines 1-7); Fig. 33, Fig. 37 and Fig. 39-41].

Shimizu, Newman, and Rutledge are analogous art because all three teach a

method of configuring hypermedia.

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to have combined the teachings of Rutledge with the combined

teachings of Shimizu and Newman because Rutledge's teachings provide the

combination of Shimizu's and Newman's method with sophisticated media search tools

to facilitate the media search process **[(page 36, paragraph [0369], lines 1-2); Fig. 39**

- Fig. 41].

Furthermore, Shimizu, Newman, and Rutledge all do not explicitly teach, "the

property editor tool is capable of configuring...hotspot properties".

However, Efrat discloses, "the property editor tool is capable of

configuring...hotspot properties" as **[*The authoring tool 1001 may be used to define***

***hotspots in media, such as video, and link 1010 the hotspots to targets* (column 5,**

lines 2-3)].

Shimizu, Newman, Rutledge, and Efrat are analogous art because all three teach

a method of configuring hypermedia.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined the teachings of Efrat with the combined teachings of Shimizu, Newman, and Rutledge because Efrat's teachings provide the combination of Shimizu's, Newman's, and Rutledge's method with the ability to create and display hyperlinks, or hotspots, in select regions of interest in frames of a video **[column 2, lines 15-17]**.

Response to Arguments

7.   Applicant's arguments filed June 25, 2008 have been fully considered but they are not persuasive.

First, Applicant argues that regarding amended independent claim 1, Shimizu, Newman, Rutledge, and Efrat do not teach, "generating at least one program that includes one or more program properties, wherein the one or more program properties are associated with the imported media and each program property is one of a common program property, a media related property, a synchronization property, an annotation property, a hotspot property, a narration property and a border property, wherein each program is a program object and each program property is a property object, and wherein each program object references at least one property object" and "editing the programs and properties based on user input obtained through the collection basket tool interface".

In response, the Examiner respectfully disagrees.

Shimizu discloses, "generating at least one program that includes one or more

program properties, wherein the one or more program properties are associated with

the imported media" as **[*The invention provides a hypermedia authoring system*

that enables an author to generate a document using a goals outline and a

presentation outline and provides a technique to relate the goals and

presentation outlines. The goals outline organizes the information content of the

document contained in cards in a logical structure while the presentation outline

is directed to a physical appearance of the document. The goals outline is

generated by instantiating document prototypes that contain slots which indicate

a type of information that is to be included in the document prototype. Each of

the slot types may be linked to the cards by links which may be mandatory,

optional, or user defined...Once instantiated, the hypermedia authoring system

provides a display that permits the author to link each of the slot types of an

instantiated document prototype with cards in a card database. Cards may...be

created from scratch by the author or imported from external sources and linked

with the instantiated document prototype (column 1, lines 43-65); The cards in the

card database 106 are information elements (called points) on which the

document being authored is based...the cards contain the actual hypermedia

information such as video, sound or text that make up the document. The

contents of the points may be appropriated from either external sources or

created from scratch by the author (column 3, lines 38-44); The hypermedia

authoring system also provides a display for creating a presentation outline in

the form of Bento-boxes. Each Bento-box includes a layout workspace in which

spacer objects may be placed. Spatial and temporal parameters may also be

specified corresponding to each spacer object. The spacer objects may be linked

to cards in the card database and an indication of a link relationship with the

goals outline is also provided (column 1, line 66 – column 2, line 6); *The related*

goals outline nodes are goals outline nodes that are linked to cards which are

also linked to spacer objects in the Bento-box 800 (column 7, lines 48-50); **Fig. 8,**

Fig. 10, Fig. 11, and Fig. 13]. This means that spacer objects (property objects) and

related goals outline nodes (property objects) are program properties of a Bento-Box

(program object) which defines, for example, a single slide (document) program. See

Fig. 13. Each Bento-box 800 includes a related goals outline area 806 that shows

related goals outline nodes. The goals outline (See Fig. 8) organizes the information

content of the document (program) contained in cards (the cards, which may be

imported from external sources, contain the actual hypermedia information such as

video, sound or text that make up the document) in a logical structure. The goals outline

is generated by instantiating document (program) prototypes that contain slots which

indicate a type of information that is to be included in the document (program)

prototype. Each of the slot types may be linked to the cards by links which may be

mandatory, optional or user defined. Each Bento-box also includes a layout workspace

in which spacer objects may be placed. See Fig. 10. Spatial and temporal parameters

may also be specified corresponding to each spacer object. See Fig. 11. Fig. 10 shows

a layout workspace 700 where the author may generate a physical appearance of the

document spatially as well as temporally. The spacer objects may be linked to cards in

the card database and an indication of a link relationship with the goals outline is also

provided. Thus, the related goals outline nodes (property objects) are linked to cards

that contain the actual hypermedia information that may be imported from external

sources (the one or more program properties are associated with the imported media)

which are also linked to spacer objects (property objects) in the Bento-box (program

object) which defines, for example, a single slide (document) program.

Furthermore, Newman discloses, "each program property is one of a common

program property, a media related property, a synchronization property, an annotation

property,...a narration property and a border property" as **[*The present invention*

provides a complete, standalone, economical system that enables consumers to

create, manipulate, edit, view and output hypermedia (column 17, lines 60-63);**

The term hypermedia refers to the integration of text, graphics, sound, video, and

other data, or any combination into a primarily associative system for information

presentation, storage and retrieval. For example, hypermedia includes motion

pictures, music, animation and photographs* (column 5, lines 47-50); *Referring

now to FIG. 8, a flowchart illustrates a method for manipulating hypermedia. At

state 380, a consumer captures hypermedia to a storyboard. The storyboard often

includes a plurality of captured hypermedia portions, or clips. At state 382, the

consumer manipulates the captured hypermedia on the storyboard...For

example, consumers manipulate clips by enhancing them with graphics, text and

audio annotations. Consumers may likewise enhance electronic mail (e-mail) by

excerpting, editing and attaching the edited clips to their e-mail messages.

Consumers also manipulate clips by compiling a sequence of clips and creating

the transitions between each pair of clips. Additionally, consumers may

manipulate clips by incorporating digitized photographs, synchronized music and

other forms of digital hypermedia captured from the Internet via the Internet

browser functionality incorporated into the non-linear editing system 100. For

example, a consumer may manipulate a clip by excerpting a video still or a digital

photograph from a clip for placement on a World Wide Web page or for printout

as a photograph or postcard. At state 384, the consumer may modify the default

transitions selected by the non-linear editing system 100. Moreover, at state 386,

the system 100 enables consumers to add overlays, such as graphical and audio

annotations of clips. Lastly, consumers may play back their storyboards at state

388 or copy their storyboards at state 390 (column 14, lines 10-37); Fig. 8].

Furthermore, Efrat discloses, "each program property is one of…a hotspot

property" as [*The authoring tool 1001 may be used to define hotspots in media,*

such as video, and link 1010 the hotspots to targets (column 5, lines 2-3)].

Furthermore, Shimizu discloses, "each program is a program object" as [*The*

cards in the card database 106 are information elements (called points) on which

the document being authored is based…the cards contain the actual hypermedia

information such as video, sound or text that make up the document. The

contents of the points may be appropriated from either external sources or

created from scratch by the author (column 3, lines 38-44); *Bento-box 800 (which*

***may be a card)* (column 7, lines 36-37); *The Bento-boxes may be linked to each**

***other in a stack for a sequential slide presentation* (column 2, lines 12-13); Fig. 13**

and Fig. 15]. This means that cards (the cards contain the actual hypermedia

information such as video, sound or text that make up the document) are program

objects. For example, a program, such as a video, may be a video card (program

object). Furthermore, a Bento-box (which may be a card) is also a program object. The

presentation outline processor provides support for the author to generate a

presentation outline (generate a scene). The Bento-box (See Fig. 13) provides a

method for linking the presentation outline to the cards in the card database (cards

contain the actual hypermedia information such as video, sound or text that make up

the document) which indirectly also links the presentation outline with the goals outline

as indicated by the related goals outline area. See 806 in Fig. 13. Thus, for example, a

single Bento Box (program object) could define a single slide or scene (program). The

Bento-boxes may be linked to each other in a stack (program object) to create a

sequential slide presentation (program). See Fig. 15.

        Furthermore, Shimizu discloses, "each program property is a property object" as

[***The spacer objects 706-718* (column 7, line 35); *related goals outline nodes**

(column 7, lines 46-48); *The related goals outline nodes are goals outline nodes

***that are linked to cards which are also linked to spacer objects in the Bento-box**

***800* (column 7, lines 48-50); *The goals outline organizes the information content**

***of the document contained in cards in a logical structure...The goals outline is**

***generated by instantiating document prototypes that contain slots which indicate**

a type of information that is to be included in the document prototype. Each of

the slot types may be linked to the cards by links which may be mandatory,

optional, or user defined **(column 1, lines 45-55);** *Each Bento-box includes a*

layout workspace in which spacer objects may be placed. Spatial and temporal

parameters may also be specified corresponding to each spacer object. The

spacer objects may be linked to cards in the card database and an indication of a

link relationship with the goals outline is also provided **(column 2, lines 1-6); Fig.**

8, Fig. 10, Fig. 11, and Fig. 13]. This means that spacer objects (property objects) and

related goals outline nodes (property objects) are program properties of a Bento-Box

(program object) which defines, for example, a single slide (document) program. See

Fig. 13. Each Bento-box 800 includes a related goals outline area 806 that shows

related goals outline nodes. The goals outline (See Fig. 8) organizes the information

content of the document (program) contained in cards (the cards contain the actual

hypermedia information such as video, sound or text that make up the document) in a

logical structure. The goals outline is generated by instantiating document (program)

prototypes that contain slots which indicate a type of information that is to be included in

the document (program) prototype. Each of the slot types may be linked to the cards by

links which may be mandatory, optional or user defined. Each Bento-box also includes a

layout workspace in which spacer objects may be placed. See Fig. 10. Spatial and

temporal parameters may also be specified corresponding to each spacer object. See

Fig. 11. Fig. 10 shows a layout workspace 700 where the author may generate a

physical appearance of the document spatially as well as temporally. The spacer

objects may be linked to cards in the card database and an indication of a link

relationship with the goals outline is also provided.

Furthermore, Shimizu discloses, "each program object references at least one

property object" as [*The related goals outline nodes are goals outline nodes that*

are linked to cards which are also linked to spacer objects in the Bento-box 800

(column 7, lines 48-50); *The cards in the card database 106 are information*

elements (called points) on which the document being authored is based...the

cards contain the actual hypermedia information such as video, sound or text

that make up the document. The contents of the points may be appropriated from

either external sources or created from scratch by the author **(column 3, lines 38-**

44); Fig. 8 and Fig. 13]. This means that spacer objects (property objects) and related

goals outline nodes (property objects) are program properties of a Bento-Box (program

object) which defines, for example, a single slide (document) program. See Fig. 13. The

related goals outline nodes (property objects) are linked to cards that contain the actual

hypermedia information which are also linked to spacer objects (property objects) in the

Bento-box (program object).

Finally, Shimizu discloses, "editing the programs and properties based on user

input obtained through the collection basket tool interface" as [*The hypermedia*

authoring system...provides a display for creating a presentation outline in the

form of Bento-boxes. Each Bento-box includes a layout workspace in which

spacer objects may be placed. Spatial and temporal parameters may also be

specified corresponding to each spacer object. The spacer objects may be linked

to cards in the card database and an indication of a link relationship with the

goals outline is also provided (column 1, line 66 – column 2, line 6); *the Bento-*

box 800 also includes a related goals outline area 806 that shows related goals

outline nodes (column 7, lines 45-47); *The goals outline organizes the information*

content of the document contained in cards in a logical structure...The goals

outline is generated by instantiating document prototypes that contain slots

which indicate a type of information that is to be included in the document

prototype. Each of the slot types may be linked to the cards by links which may

be mandatory, optional, or user defined (column 1, lines 45-55); *The related goals*

outline nodes are goals outline nodes that are linked to cards which are also

linked to spacer objects in the Bento-box 800 (column 7, lines 48-50); **Fig. 8, Fig.**

10, Fig. 11, Fig. 13, and Fig. 15]. This means that the hypermedia authoring system

provides a display for creating a presentation outline (document) in the form of Bento-

boxes. Spacer objects (property objects) and related goals outline nodes (property

objects) are program properties of a Bento-Box (program object) which defines, for

example, a single slide (document) program. See Fig. 13. The Bento-boxes may be

linked to each other in a stack (program object) to create a sequential slide presentation

(program) See Fig. 15. Each Bento-box 800 includes a related goals outline area 806

that shows related goals outline nodes. The goals outline (See Fig. 8) organizes the

information content of the document (program) contained in cards (the cards contain the

actual hypermedia information such as video, sound or text that make up the document)

in a logical structure. The goals outline is generated by instantiating document

(program) prototypes that contain slots which indicate a type of information that is to be included in the document (program) prototype. Each of the slot types may be linked to the cards by links which may be mandatory, optional or user defined. Each Bento-box also includes a layout workspace in which spacer objects may be placed. See Fig. 10. Spatial and temporal parameters may also be specified corresponding to each spacer object. See Fig. 11. Fig. 10 shows a layout workspace 700 where the author may generate a physical appearance of the document spatially as well as temporally. The spacer objects may be linked to cards in the card database and an indication of a link relationship with the goals outline is also provided.

Second, independent claims 31 and 32 and dependent claims 3-4, 6, 8-12, and 19-30 recite similar limitations argued by the Applicants and are therefore also rejected for the same above reasons.

Conclusion

8.   Applicant's amendment necessitated the new ground(s) of rejection presented in

this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP

§ 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37

CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action. In the event a first reply is filed within

TWO MONTHS of the mailing date of this final action and the advisory action is not

mailed until after the end of the THREE-MONTH shortened statutory period, then the

shortened statutory period will expire on the date the advisory action is mailed, and any

extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later

than SIX MONTHS from the date of this final action.

Contact Information

9.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Michele C. Choi whose telephone number is 571-272-

9776. The examiner can normally be reached on Mon-Fri, 7:30AM to 5PM EST,

alternate Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Tony Mahmoudi can be reached on 571-272-4078. The fax phone number

for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for published

applications may be obtained from either Private (PAIR) or Public PAIR. Status

information for unpublished applications is available through Private PAIR only. For

more information about the PAIR system, see http://pair-direct.uspto.gov. Should you

have questions on access to the Private PAIR system, contact the Electronic Business

Center (EBC) at 866-217-9197 (toll free). If you would like assistance from a USPTO

Customer Service Representative or access to the automated information system, call

800-786-9199 (IN USA OR CANADA) or 571-272-1000.

September 19, 2008   Michele C. Choi
                                                          /M. C. C./
                                                          Examiner, Art Unit 2169

/S. P./
Primary Examiner, Art Unit 2164

/Tony Mahmoudi/

Supervisory Patent Examiner, Art Unit 2169